

Article

Application of Norm Optimal Iterative Learning Control to Quadrotor Unmanned Aerial Vehicle for Monitoring Overhead Power System

Husam A. Foudeh ^{1,*} , Patrick Luk ^{1,*}  and James Whidborne ² ¹ Electric Power and Drives Group, Cranfield University, Cranfield MK43 0AL, UK² Centre for Aeronautics, Cranfield University, Cranfield MK43 0AL, UK; j.f.whidborne@cranfield.ac.uk

* Correspondence: Husam-Abdel-Rasoul-Yousef.Foudeh@cranfield.ac.uk (H.A.F.); p.c.k.luk@cranfield.ac.uk (P.L.); Tel.: +44-1234-75-4716 (P.L.)

Received: 15 March 2020; Accepted: 8 June 2020; Published: 22 June 2020



Abstract: Wind disturbances and noise severely affect Unmanned Aerial Vehicles (UAV) when monitoring and finding faults in overhead power lines. Accordingly, we propose repetitive learning as a new solution for the problem. In particular, the performance of Iterative Learning Control (ILC) that are based on optimal approaches are examined, namely (i) Gradient-based ILC and (ii) Norm Optimal ILC. When considering the repetitive nature of fault-finding tasks for electrical overhead power lines, this study develops, implements and evaluates optimal ILC algorithms for a UAV model. Moreover, we suggest attempting a learning gain variation on the standard optimal algorithms instead of heuristically selecting from the previous range. The results of both simulations and experiments of gradient-based norm optimal control reveal that the proposed ILC algorithm has not only contributed to good trajectory tracking, but also good convergence speed and the ability to cope with exogenous disturbances such as wind gusts.

Keywords: unmanned aerial vehicles (UAVs); quadrotor; Iterative Learning Control (ILC); Norm Optimal ILC; gradient-based ILC; power system; inspection task

1. Introduction

Overhead electrical power lines are a vital component of the power supply infrastructure. It is essential to carry out preventive monitoring of high voltage transmission lines more safely and efficiently in order to meet consumer demand [1–3]. UAVs have the potential to perform these monitoring tasks, due to their inherent advantages of cost, manoeuvrability, speed, and easy set-up [4,5]. They can attain the required heights and positions that are needed for performing major inspection tasks and do not require contact. However, UAVs are susceptible to aerodynamic disturbances [6,7]. Because of this, the autonomous identification of faults in terms of performance and accuracy for overhead power lines remains a difficult problem. To ensure the quality of the tasks in fault-finding operations for overhead power lines, precise and accurate trajectory tracking is important [8,9].

The UAV performance relies both on the chosen control strategy and on the underlying vehicle dynamics. Quadrotors are under-actuated and open-loop unstable with significant non-linearity and strong dynamic coupling. Furthermore, exogenous disturbances can easily compromise quadrotor stability [10]. These make the control problem non-trivial. To design and test the control system, models of the dynamics are required. However, the system is subject to modelling, plant and disturbance uncertainties that should be accounted for.

Researchers have developed an ongoing design capability to overcome limitations using various control methods. Some depend on classical control (e.g., PID controller), which is simple and capable of providing acceptable performance [11]. Nevertheless, this method needs an accurate mathematical

model, as do Linear Quadratic Regulator (LQR) [12] approaches. Others are based on non-linear control such as sliding-mode [13] or back-stepping [14]. These works make use of conventional feedback and feedforward and to improve the control performance by augmenting the previous controllers (i.e., PID controller). Consequently, the feedback controllers react on the basis of the observation of information using reference signal and disturbance that results in a delayed tracking response [15]. In these regimes, it can be observed that feedback is unable to react in time.

To achieve high performance, extending classical control methods with learning schemes is another approach. Iterative Learning Control (ILC) algorithms provide some advantages over conventional feedback controllers in that they develop an element of intelligence by memorizing from previous practice [16]. ILC has been applied to quadrotors using various approaches. These include derivative-type ILC [17], Proportional-Interactive-Derivative (PID-type) ILC [18], and basic optimal ILC approaches [19] that assume the reference (r) is specified over the entire finite time horizon, $[0, T]$, and they also do not require a model. On the other hand, ILC has been applied to some complicated systems, such as precise speed-tracking control of a robotic fish [20], for multi-agent systems in [21], and recently in [22] for the distributed ILC of multiple flexible manipulators in the presence of uncertain disturbances and actuator dead zones.

We propose using two ILC methods that are based on optimal approaches. These are the Gradient-based ILC and the Norm Optimal ILC. They are chosen because of their prevalence, their use of the same underlying cost function, and because they have similar implementations [23]. This paper seeks to determine whether the proposed optimal algorithms can be applied to quadrotors for tracking performance. Furthermore, it proceeds to motivate, develop, and evaluate the flight controller required to address and if it can address this deficiency of these previous methods.

Existing ILC approaches to UAV control are reviewed in the next section. The ILC design, application, and optimal algorithm designs are described in Section 3. A suitable experimental system is selected in order to provide a test platform and results for the proposed algorithm G-ILC and NO-ILC presented in Sections 5 and 4, respectively. Section 6 provides some concluding remarks.

2. ILC Controllers

2.1. Basic ILC

ILC relies on performing similar missions multiple times, so that the control can be modified to improve performance over previous operations (i.e., trials, iterations, and passes) through learning. However non-learning systems do not improve their performance due to the same tracking error on each iteration in which despite large model uncertainty and repeating disturbances [23]. Learning-type control strategies can accordingly be classified into ILC, Repetitive Control (RC), neural networks, and adaptive control. Whilst ILC strategies modify the input signal (i.e., the control input), adaptive and neural network learning control methods modify the system (i.e., the controller), and controller parameters, respectively [24]. Additionally, ILC usually guarantees fast convergence within just a few iterations, but the alternative strategies may not [25].

ILC has been applied to quadrotors in a few cases. ILC can be used for systems for which a finite-duration task is repeated. Every iteration should have the same initial conditions and, as the number of trials increase, ILC updates the input signal to ensure that the system output converges to a reference signal. ILC has been applied to many fields, including robotics [26].

The simple D-type ILC form used in [27] for UAV trajectory tracking was based on the Additive State Decomposition (ASD) method. The block diagram shown in Figure 1 demonstrates the D-type algorithm, which employs the change of error rate to modify the input for the next iteration in lieu of the error itself. It is evident that the derivative part in the ILC algorithm amplifies small noise signals which may destabilize the system. Ref. [28] has developed another special case of D-type in UAV applications in order to obtain a better tracking performance and lower errors than the typical D-type update rule in [27]. However, although both [27] and [28] use the D-type ILC, they still do not guarantee the rate of convergence in the presence of disturbances.

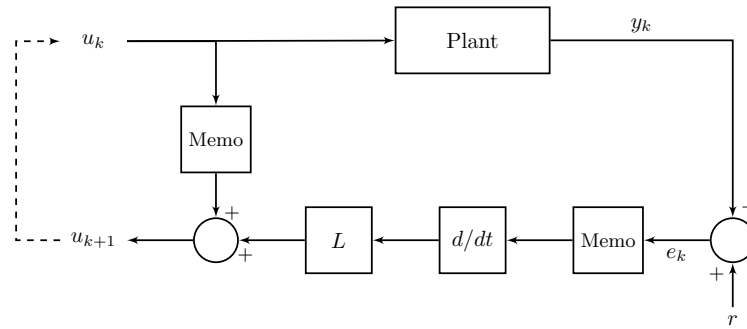


Figure 1. Block diagram for D-type ILC:- here the $u_k(n)$ it is the input signal was used on k th iterations, L_{opt} is the D-type gain, $e_k(n)$ the derivative of error, and $r(n)$ and $y_k(n)$ are the reference and the plant output, respectively.

In [29,30], the P and D-term ILCs are combined as a PD-type ILC to increase the convergence rate. In [29], three different methods are additionally applied: offline ILC, online ILC, and a combination of both ILCs. These have the respective forms for P-type ILC

$$u_{k+1}(n) = \underbrace{u_k(n) + K_P e_k(n)}_{\text{offline P-type}}, \quad (1)$$

$$u_{k+1}(n) = \underbrace{u_k(n) + K_P e_{k+1}(n)}_{\text{online P-type}}, \quad (2)$$

where subscript k denotes the iteration number and subscript n the the sample number.

An inner online PD type ILC update was designed by [29] for quadrotor trajectory tracking control to stabilise the UAV system without taking disturbances into account. The algorithm is

$$u_{k+1}(n) = u_k(n) + K_P e_k(n+1) + K_D [e_k(n+1) - e_k(n)] \quad (3)$$

The results show a high tracking error. However, the ILC was able to reduce it over subsequent iterations but with low convergence. In [30], ILC was implemented by an adaptive term for enhancing the performance and robustness. This controller term was implemented on a quadrotor platform, where the test results showed improvement in tracking performance, despite the presence of disturbances.

In [31], another ILC form is proposed that includes a combination of the previous two simple structures to include an integral term and is termed PID-type ILC. A controllable flight was optimized using the PID-type ILC after a chnge in mass of the quadrotor. The method is based only on manual auto-tuning for parameters. In summary, all the previous structures (D-type, PD type, PID-type ILC) are susceptible to process disturbance and measurement error, while rarely being utilized in practical applications.

ILC was applied to achieve quadrotor trajectory tracking while balancing an inverted pendulum [32]. The learning algorithm used was of the form

$$\begin{cases} \min_{u_{k+1}} \|S(Fu_{k+1} + \hat{d}_k)\| + \alpha \|Du_{k+1}\| \\ \text{subject to } L_{opt} u_{k+1} \leq q_{max}, \end{cases} \quad (4)$$

where F is the lifted system matrix, α weights the additional penalty term, and \hat{d}_k is an updated estimate of disturbance. Via the matrix D , the input derivatives can be penalized. The matrix S allows for the error signal to be scaled or filtered.

The aforementioned approaches are very limited in accuracy. Apart from initial identification procedures and tuning, it is also noted that these approaches demand a large level of computation and do not require an explicit model. Although usability is an advantage of this simplicity, it necessarily

degrades performance. There is a great opportunity to assess a wide variety of ILC approaches on UAVs. There is no single algorithm that delivers all of the required features for high performance control while facing uncertain dynamics and environmental factors. Overall, ILC approaches demonstrated the best tracking performance only with medium complexity. Relatively few ILC schemes have been applied to quadrotors, and their evaluation is quite limited.

2.2. Optimal Approach ILCs

The properties of linear optimal algorithms have been studied extensively [33–37]. Leading ILC examples are now introduced with their own specific features.

2.2.1. Gradient-Based Iterative Learning Control

Due to their attractive theoretical properties, Gradient-based (G-ILC) algorithms have received considerable attention in the literature. When compared to generic ILC approaches, the optimal gradient ILC achieves faster error convergence by relying on the system model and utilizing the characteristics of gradient-descent in order to structure the ILC control action update. In [38–40], Gradient ILC has been used for SISO systems and was derived for MIMO systems.

The common form of standard ILC employed by the input update law is specified as [23]

$$u_{k+1}(n) = u_k(n) + L_{opt}e_k(n) \quad (5)$$

where L_{opt} is a learning operator, y_d is the desired reference signal, and $e_k = y_d - y_k$ is the error. By taking the transpose, G^T , an alternative method of guaranteeing Equation (5) can be obtained. The lifted form of the more general adjoint operator G^* leads to

$$L_{opt} = \beta G^* \quad (6)$$

which yields the update law

$$u_{k+1} = u_k + \beta G^T e_k \quad (7)$$

where the scalar β is the learning gain. Equation (7) is interpreted to be the gradient descent solution to the minimisation problem $\min_u J(u_k) = \|y_d - Gu_k\|^2$. The spectral radius condition is a necessary and sufficient condition for convergence

$$\rho(I - GL_{opt}) < 1. \quad (8)$$

Substituting $L_{opt} = \beta G^T$ into the general convergence conditions Equation (8) yields

$$\rho(I - \beta GG^T) = \|I - \beta GG^T\| < 1 \quad (9)$$

Here, the GG^T is positive definite, so the convergence condition becomes

$$0 < \beta < \frac{2}{\|GG^T\|} = \frac{2}{\|G\|^2}. \quad (10)$$

Therefore, it should be noted, when the number of trials, k , approaches infinity, Equation (10) ensures the error converges monotonically to zero.

2.2.2. Norm Optimal ILC

The model-based Norm Optimal ILC (NO-ILC) algorithm was introduced in [41]. The ILC input to the following trial is acquired through optimising a specific performance index that allows a balance between the error convergence and input energy. NO-ILC has been used for many applications and

extended, for instance a predictive approach based on norm-optimal ILC [42]. NO-ILC uses the quadratic cost function:

$$J(u_{k+1}) = \frac{1}{2} \{ [u_{k+1} - u_k]^T R [u_{k+1} - u_k] + [y_d - y_{k+1}]^T Q [y_d - y_{k+1}] \} \quad (11)$$

where the weighting matrices $R(t)$ and $Q(t)$ are positive semi-definite and symmetric.

The requirement is to minimize the tracking error by modifying the input control from one trial pass to the next. This can be done by generating the control action u_{k+1} for the next trial. The problem at each iteration is thus

$$\min J(u_{k+1}) \quad (12)$$

which can be solved by applying a partial differentiation to J_{k+1} with respect to u , and determining the stationary point, $\partial J_{k+1} / \partial u = 0$. This leads to the update law

$$u_{k+1} = u_k + G^T e_{k+1} \quad (13)$$

where G^T is the adjoint operator of the system G , where $G^T = R^{-1} G^T Q$. It can be demonstrated that the convergence condition in Equation (8) is always satisfied and, moreover,

$$\|e_{k+1}\| \leq \frac{1}{1 + \sigma^2(GG^T)} \|e_k\| \leq \|e_k\| \quad (14)$$

where $\sigma^2(GG^T)$ is the smallest spectral radius value of the symmetric, positive definite operator GG^T . Equation (14), being non-causal, can be manipulated to either generate a causal feed-forward form or feedback and feed-forward form by [43], where

$$L_{opt} = G^T (I - GG^T)^{-1}. \quad (15)$$

Therefore, the updated law is

$$u_{k+1} = u_k + G^T (I - GG^T)^{-1} e_k. \quad (16)$$

Note that the implementation of optimal algorithms in such cases needs to be investigated. This is more crucial in the application of UAVs for the monitoring overhead power system, especially since the quadrotor UAV has more than one degree of freedom. Moreover, the electric power system inspection task is inherently repetitive while detecting errors that require using optimal algorithms to critically compare performance and inform design.

3. ILC Design and Application to Quadrotor

This section purports to put forward the optimal algorithms (G-ILC, NO-ILC) for the UAV quadrotor. The design of the optimal algorithm is based on the following assumptions and steps :

- I. The system is presumed to operate in a repetitive manner (iteratively) for both optimal algorithms, G-ILC and NO-ILC.
- II. At the end of every iteration, the state is reset operation toward a particular repetition that have independent initial condition to the next operation.
- III. A new control signal might be utilized during this time. A reference signal, $r(t)$, is presumed to be known and the ultimate control objective is to determine an input function $u^*(t)$ such the output function $y(t) = r(t)$ on $[1; N]$.
- IV. For G-ILC, the value of the learning gain $\beta_{old}(k)$ is heuristically selected for the first step, and then calculated automatically using the the gain $\beta_{new}(k)$ by establishing the varying gain equations. The established variable will be repeated again for NO-ILC, but with a different learning gain Q_k .
- V. To guarantee error convergence, the necessary conditions are $J(\beta_{new}(k)) = \|e_{k+1}\|^2 + \zeta \beta_{new}(k)^2$.

Now, the SISO model is a non-linear, discrete, state space system:

$$\begin{aligned}x_k(n+1) &= f(x_k(n), u_k(n)) \\y_k(n) &= h(x_k(n))\end{aligned}\quad (17)$$

where $x_k(t) \in \mathbb{R}^n$, $u_k(t) \in \mathbb{R}^m$, $y_k(t) \in \mathbb{R}^p$ and $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$ are the system matrices. Moreover, x_k , u_k , and y_k are the state vector, input and output respectively, for trial k .

3.1. Gradient-Based ILC (G-ILC)

When compared to generic ILC approaches, the optimal gradient ILC depends on the system model to obtain faster convergence. It constructs the ILC control action update while using the properties of gradient descent. This happens through minimising the cost function

$$\min J(u_k) = \frac{1}{2} \|e_k\|^2 = \frac{1}{2} \|y_d - Gu_k\|^2 \quad (18)$$

where

$$G = \begin{bmatrix} g_0 & 0 & \cdots & 0 \\ g_1 & g_0 & \cdots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ g_{h-1} & g_{h-2} & \cdots & g_0 \end{bmatrix} \quad (19)$$

$$g_N = CA^{N+r-1}B \quad N = 0, 1, 2, \dots, h-1$$

and the tracking error e_k from the N th trial. This is the error between the actual outputs y_k of the system and their desired reference signal y_d is then

$$e_k = y_d - y_k. \quad (20)$$

Using gradient descent to solve the optimisation problem given by Equation (18) gives

$$u_{k+1} = u_k - \beta \nabla J(u_k) \quad (21)$$

$$= u_k - \beta \nabla \frac{1}{2} \|y_d - Gu_k\|^2 \quad (22)$$

$$= Gu_k\beta + u_k - y_d G^T \beta \quad (23)$$

$$= u_k + \beta G^T e_k \quad (24)$$

where β represents the learning gain.

From Equation (24), the error evolution of the G-ILC can be derived as

$$e_{k+1} = y_d - Gu_{k+1} \quad (25)$$

$$= G^T Ge_k\beta + y_d - u_k G \quad (26)$$

$$= (I - \beta GG^T)e_k. \quad (27)$$

By choosing the learning gain β from the range $0 < \beta < 2/\bar{\sigma}(G)$, where $\bar{\sigma}(G)$ is the largest singular of the matrix G , it can be easily shown that $\|I - \beta GG^T\| < 1$. Therefore, the error converges monotonically to zero, as the trials k goes to infinity.

Instead of arbitrarily selecting a value of $\beta_{old}(k)$ from the range, the error convergence rate can be optimized. Repeating Equations (23) and (27)

$$u_{k+1} = u_k + \beta_{old}(k) G^T e_k \quad (28)$$

$$e_{k+1} = (I - \beta_{old}(k) GG^T)e_k \quad (29)$$

the optimal iteration-varying $\beta_{new}(k)$ can be obtained by minimising:

$$J(\beta_{new}(k)) = \|e_{k+1}\|^2 + \zeta\beta_k^2. \quad (30)$$

where ζ is a small positive weighting constant. Substituting Equation (29) into Equation (30) we get

$$J(\beta_{new}(k)) = ((I - \beta_k GG^T)e_k)^T((I - \beta_k GG^T)e_k) + \zeta\beta_k^2 \quad (31)$$

$$= e_k^T e_k - 2\beta_k e_k^T GG^T e_k + \beta_k^2 e_k^T GG^T GG^T e_k + \zeta\beta_k^2. \quad (32)$$

Differentiating Equation (32) with respect to $\beta_{new}(k)$ and equating to zero gives the optimal learning gain:

$$\beta_{new}(k) = \frac{e_k^T GG^T e_k}{e_k^T GG^T GG^T e_k + \zeta} \quad (33)$$

$$= \frac{(G^T e_k)^T G^T e_k}{(GG^T e_k)^T GG^T e_k + \zeta} \quad (34)$$

$$= \frac{\|G^T e_k\|^2}{\|GG^T e_k\|^2 + \zeta}. \quad (35)$$

Thus the necessary and sufficient conditions for guaranteeing a convergence of error are

$$\|e_{k+1}\| < \|e_k\| \text{ for all } k \geq 0 \text{ and } \lim_{k \rightarrow \infty} e_k = 0. \quad (36)$$

From Equation (29) we get

$$\|e_{k+1}\|^2 - \|e_k\|^2 = e_k^T e_k (I - \beta_k GG^T)(I - \beta_k GG^T)^T - e_k^T e_k \quad (37)$$

$$= e_k^T ((I - \beta_k GG^T)^2 - I) e_k \quad (38)$$

$$= e_k^T (-2\beta_k GG^T + \beta_k^2 GG^T GG^T) e_k \quad (39)$$

$$= \beta_k^2 (-2 \frac{e_k^T GG^T e_k}{\beta_k} + e_k^T GG^T GG^T e_k) \quad (40)$$

$$= \beta_k^2 (-2 \frac{\|G^T e_k\|^2}{\beta_k} + \|GG^T e_k\|^2). \quad (41)$$

Furthermore from Equation (34), we get

$$\frac{\|G^T e_k\|^2}{\beta_{new}(k)} = \|GG^T e_k\|^2 + \zeta. \quad (42)$$

Substituting Equation (41) into Equation (42) gives

$$\begin{aligned} \|e_{k+1}\|^2 - \|e_k\|^2 &= \beta_{new}(k)^2 (-2(\|GG^T e_k\|^2 + \zeta) + \|GG^T e_k\|^2) \\ &= -\beta_{new}(k)^2 (2\zeta + \|GG^T e_k\|^2) \leq 0. \end{aligned} \quad (43)$$

From Equation (43) it can be deduced that $\|e_{k+1}\| = \|e_k\|$ if and only if $\beta_k = 0$. Because GG^T is a positive definite matrix, from Equation (34) we have that $\beta_{new}(k) = 0$ if and only if $e_k = 0$. Thus the conditions of Equation (36) are satisfied and the system has monotonic convergence.

3.2. Norm Optimal ILC (NO-ILC)

To produce the optimal action ILC \mathbf{u}_{k+1} , we recall the Equation (11). Setting the gradient to zero gives

$$\nabla J(\mathbf{u}_{k+1}) = -G^T Q e_{k+1} + R(\mathbf{u}_{k+1} - \mathbf{u}_k) = 0. \quad (44)$$

Since matrix R is a positive definite so is non-singular, rearranging gives

$$\mathbf{u}_{k+1} = \mathbf{u}_k + R^{-1}G^T Q \mathbf{e}_{k+1}. \quad (45)$$

From Equation (45), the error evolves as

$$\mathbf{e}_{k+1} = \mathbf{y}_d - G\mathbf{u}_{k+1} \quad (46)$$

$$= \mathbf{y}_d - R^{-1}GG^T Q - \mathbf{u}_k G_k \mathbf{e}_{k+1} \quad (47)$$

$$= (I - GR^{-1}G^T Q)^{-1} \mathbf{e}_k \quad (48)$$

However Equation (11) is implicit. However, it can be solved by supposing that $G^* = R^{-1}G^T Q$ and substituting the Equation (48) into Equation (11) gives

$$\mathbf{u}_{k+1} = \mathbf{u}_k + G^* \mathbf{e}_k (I - GG^*)^{-1}. \quad (49)$$

Monotonic convergence can be shown as follows

$$\lim_{k \rightarrow \infty} \|\mathbf{e}_k\|^2 = \lim_{k \rightarrow \infty} \|\mathbf{J}_k\|^2 : J_\infty \geq 0, \quad (50)$$

implying

$$\lim_{k \rightarrow \infty} \|u_{k+1} - u_k\|^2 = 0. \quad (51)$$

Thus

$$\lim_{k \rightarrow \infty} (u_{k+1} - u_k) = \lim_{k \rightarrow \infty} R^{-1}G^T Q \mathbf{e}_{k+1} = 0. \quad (52)$$

We get error convergence if either there exist no e such that $G^T e = 0$ so $\lim_{k \rightarrow \infty} e_{k+1} = 0$, or if $\mathbf{y}_d \in \text{range}(G)$.

3.3. Application to Quadrotors

The dynamics of standard quadrotors are well established, the main equations are given here. Details of the dynamics model can be found in [34], for example.

The control inputs are related to each rotor speed Ω_i by:

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & lb & 0 & -lb \\ lb & 0 & -lb & 0 \\ d & d & d & d \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \quad (53)$$

where and l is the arm length and b and d are rotor thrust and drag coefficients respectively.

The dynamic model for the quadrotor attitude is given by

$$\dot{p} = \frac{1}{I_{xx}} [u_2 + qr(I_{yy} - I_{zz}) + qJ_p\Omega_r] \quad (54)$$

$$\dot{q} = \frac{1}{I_{yy}} [u_3 + pr(I_{zz} - I_{xx}) + pJ_p\Omega_r] \quad (55)$$

$$\dot{r} = \frac{1}{I_{zz}} [u_4 + qp(I_{xx} - I_{yy})] \quad (56)$$

where the triplet (p, q, r) are the rotation rates about the body axes, I_{xx} , I_{yy} , I_{zz} are the moments of inertia about the body axes, and J_p is the rotor moment of inertia about the rotor rotation axis.

We define a state variable vector as

$$\mathbf{x} = [\phi \quad \theta \quad \psi \quad x \quad y \quad z \quad \dot{\phi} \quad \dot{\theta} \quad \dot{\psi} \quad \dot{x} \quad \dot{y} \quad \dot{z}]^T \quad (57)$$

where the triplet (x, y, z) is the position of the vehicle in the earth axes, and (ϕ, θ, ψ) are the standard aerospace Euler angles. By approximating the rotation rate triplet (p, q, r) by the Euler angle derivative $(\dot{\phi}, \dot{\theta}, \dot{\psi})$ and from the standard aeronautics navigation equations we get the dynamic model in the form $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$ where

$$f(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \dot{\phi} \\ \dot{\theta}\psi a_1 + \dot{\theta}a_2\bar{\Omega} + b_1u_2 \\ \dot{\theta} \\ \dot{\phi}\psi a_3 - \dot{\phi}a_4\bar{\Omega} + b_2u_3 \\ \dot{\psi} \\ \dot{\theta}\phi a_5 + b_3u_4 \\ \dot{x} \\ \frac{1}{m}(\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi)u_1 \\ \dot{y} \\ \frac{1}{m}(\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi)u_1 \\ \dot{z} \\ g - \frac{1}{m}(\cos\phi \cos\theta)u_1 \end{bmatrix},$$

and where m is the mass, g is the gravitational constant, $\bar{\Omega}$ is the rotor rotation rate sum, $a_1 = (I_{yy} - I_{zz})/I_{xx}$, $a_2 = J_p/I_{xx}$, $a_3 = (I_{zz} - I_{xx})/I_{yy}$, $a_4 = J_p/I_{yy}$, $a_5 = (I_{xx} - I_{yy})/I_{zz}$, $b_1 = l/I_{xx}$, $b_2 = l/I_{yy}$, and $b_3 = l/I_{zz}$.

The SISO structure of Equation (17) is extended to a MIMO dynamics to give

$$\begin{aligned} x_k(n+1) &= f(x_k(n), u_k(n)), \\ y_k(n) &= h(x_k(n), u_k(n)), \quad x(0) = x_0. \end{aligned} \quad (58)$$

The model $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$ can be discretized by an Euler approximation. Full state feedback is assumed, that is $y_k = x_k$.

4. Experimental Platform Selection

4.1. Physical Parameters

The AscTec Hummingbird is chose as the experimental test platform. This quadrotor is popular, has good performance and is light-weight maneuverable. It has a payload of 200 g and a flight endurance of nearly 20 min. The aircraft component frame is made out of balsa wood and carbon fiber. The vehicle is powered by four brushless DC motors running off an 11.1V Lithium Polymer (LiPo) battery pack. It is equipped with an accelerometer, pressure sensor, magnetic sensor, gyros, and GPS module. These can provide the vehicle state. Some of the technical details are listed in Table 1 [44]. The model parameters are given in Table 2.

4.2. Test Bed

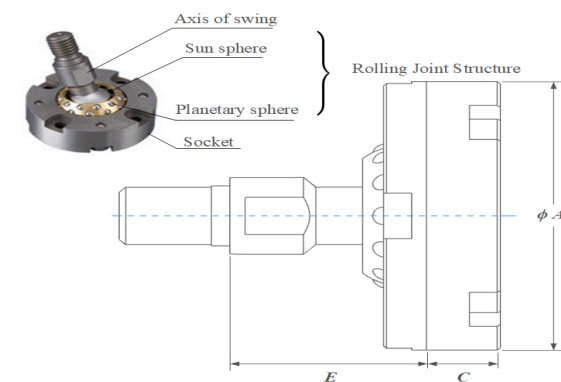
A test bed, designed for analysing the motor's performance and enabling controller tuning, is constructed from steel and finished in black paint and bearings, so that it allows three DOF of rotation. Steel tube was selected because of its easy availability and high density gives the rig stability and rigidity. The UAV is secured in place with a spherical rolling joint. The assembled mechanical design is shown in Figures 2 and 3, The UAV installed on the top. A Raspberry Pi 3 is used for the control.

Table 1. Technical Details of the Quadrotor.

Type of Platform	AscTec HB. UAV
The Producer	AscTec GmbH + Intel
Take-off Weight	480 g
Battery	2100 mAh LiPo
Distance between motors	34 cm
Propeller	Standard propellers (8"), flexible (PP) plastics
Motors	HACKER Motors Germany (X-BL 52 s)
Motor Controllers	X-BLDC controllers
Transmitter	Futaba 2.4 GHz
Wireless Link	Xbee 2.4 GHz

Table 2. Quadrotor model parameters.

Parameters	Description	Value	Unit
I_{xx}	x axis-Moment of Inertia	10.7×10^{-3}	$\text{kg} \cdot \text{m}^2$
I_{yy}	y axis-Moment of Inertia	10.7×10^{-3}	$\text{kg} \cdot \text{m}^2$
I_{zz}	z axis-Moment of Inertia	18.4×10^{-3}	$\text{kg} \cdot \text{m}^2$
J_p	Rotor Inertia	47×10^{-6}	$\text{kg} \cdot \text{m}^2$
m	Mass	0.547	kg
l	Arm Length	0.168	m
g	Gravitational constant	9.81	ms^2
Ω_{\max}	Maximum rotor speed	200	rad/s

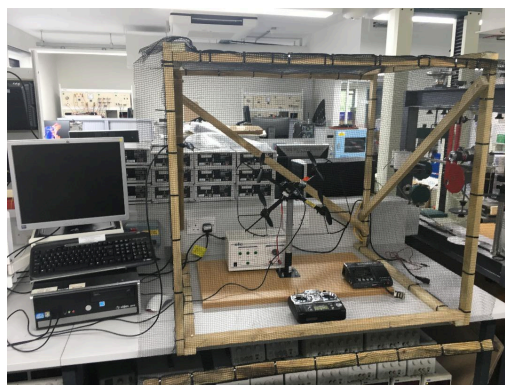


Spherical Rolling Joint SRJ Series

(a)



(b)

Figure 2. The complete assembly of proposed mechanical test-bed for quadrotor: (a) High precision spherical rolling joints with its features. (b) Mechanical-bed reconfiguration with precision rolling joints.**Figure 3.** Quadrotor system with complete hardware test frame design for Iterative Learning Control (ILC) controllers.

5. Results and Discussion

The G-ILC and NO-ILC algorithms are applied to the test system. Simulations are also performed. The simulations and experiments were conducted on a Laptop (i7) ThinkPad P1 Mobile WorkStation with 16 GB RAM/2.20 GHz via MATLAB R2018b. The reference trajectory is shown in Figure 4. The trajectory consists of a single period sin wave and is non-smooth; hence is a challenging task for the ILC algorithm. Sixteen iteration trials were performed for each algorithm. The input update for the G-ILC and NO-ILC algorithms was acquired by Equations (28) and (35), respectively, with the help of the linearized quadrotor dynamics from Equation (49).

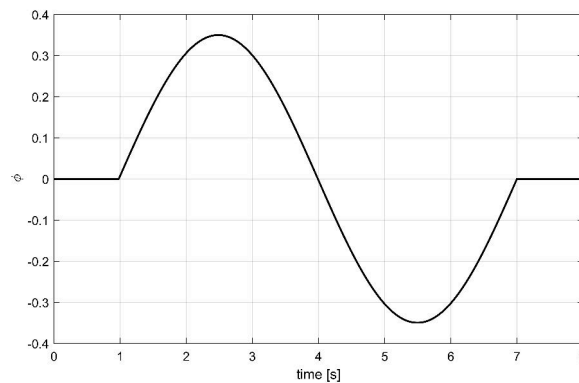


Figure 4. References for ILC controllers.

Demonstrating the monotonic convergence of the G-ILC algorithm is also important. The simulation results show a notable decrease in the error over different trial iterations. Figure 5 shows the decrease of the 2-norm of the error, with a value of 0.3092 at the 16th iteration. The variation in ϕ and θ over time for different iterations are also shown in Figure 5.

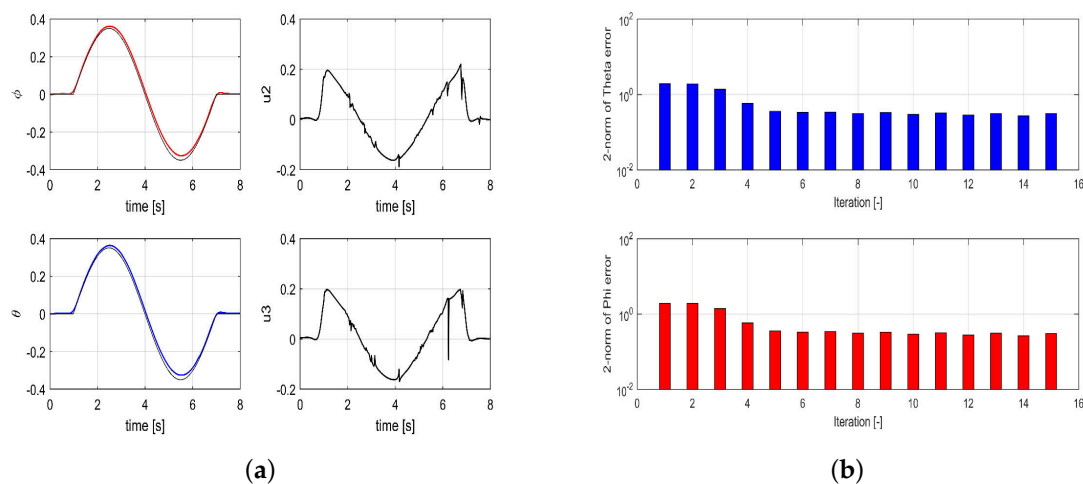


Figure 5. G-ILC for different iterations without disturbance: (a) the variation in theta over time for different iterations. (b) two norm for roll and pitch over time for different iterations.

Figure 6 similarly shows the results for the NO-ILC algorithms. The improvement over the G-ILC is noticeable. Again the convergence is shown by the decrease of the error 2-norm. The error 2-norm is 0.1215 at 16th iteration, a considerable improvement over the G-ILC approach.

The G-ILC with updating Equations (28) and (29) was implemented to track the reference signal. An optimal value of gain, β , is chosen between 0.01 and 1.0. After testing a wide range of values of β , the best performance was found with $\beta = 0.1$. The experimental results show a significant decline in

the error over the first five trial iterations as shown in Figure 7a. A slight increase occurred at the 7th and 10th trials but the trend was from 1.277 at first trial to the value of 0.574 at the 6th trial.

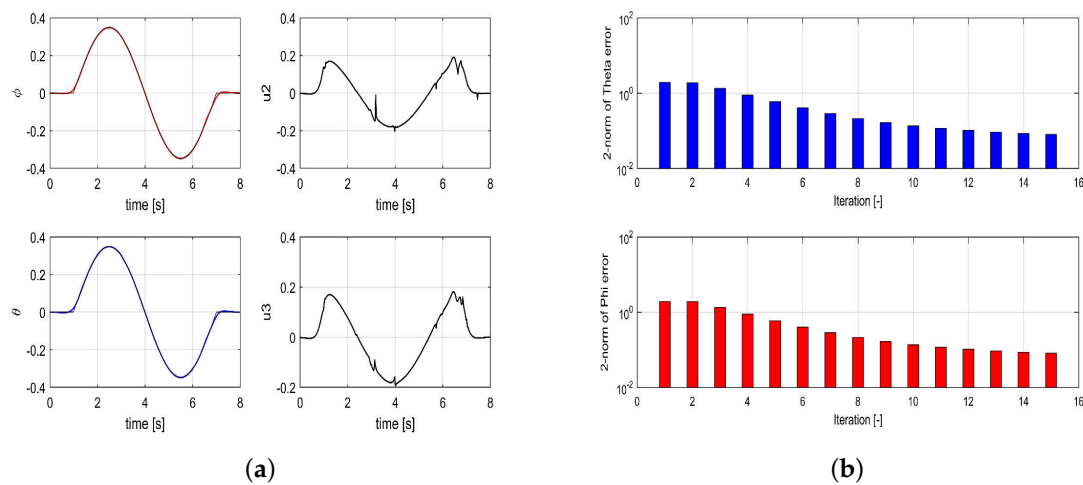


Figure 6. Norm Optimal ILC (NO-ILC) for different iterations without disturbance: (a) the variation in theta over time for different iterations. (b) two norm for roll and pitch over time for different iterations.

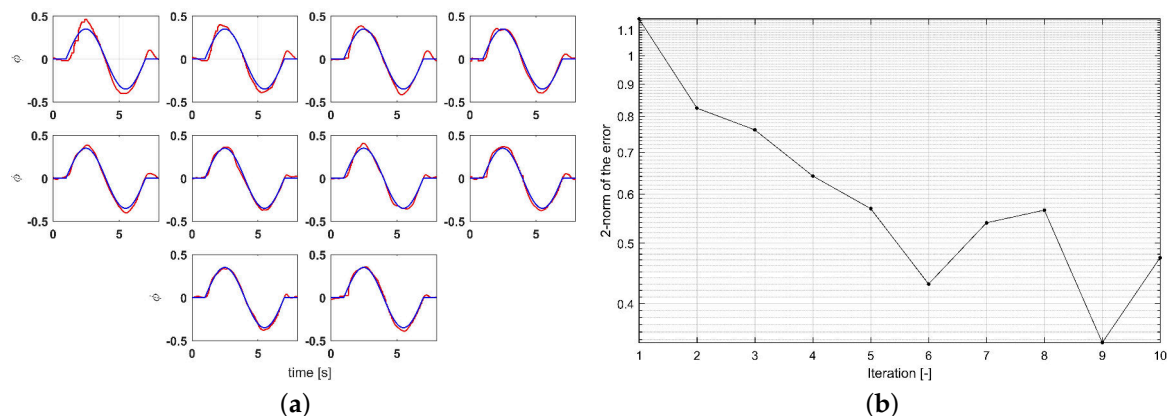


Figure 7. Experimental results of G-ILC for different iterations with disturbance. (a) Experimental tracking with different iteration. (b) Monotonic convergence result.

The performance of the NO-ILC algorithm is also investigated with the reference signal shown in Figure 4. The results are shown in Figure 8a. The weighting parameter is set to $Q = 0.1$. The value of Q can be increased to improve convergence, but Figure 8b shows the convergence is similar to that of the G-ILC experiment, with the latter slightly better. Convergence was achieved after 8 iterations.

Note that although convergence is established theoretically, in practice the system is subject to disturbances and uncertainty. The effect of disturbances is evaluated in simulation and by experiment for the two approaches. First the performances of the two methods in simulation without disturbance are quantified and compared. The results are shown in Table 3. The NO-ILC method had significantly better performance and convergence properties in simulation.

The disturbances took the form of torques that were injected in the ϕ and θ channels. The disturbances defined as exponentially decaying sinusoidal functions $\delta\tau = e^{-0.1t}(\sin t, \cos t, 0)$ for $t \in (2, 6)$ s. The results for experiment are shown in Table 4. These show the better performance of the NO-ILC but the difference is less marked.

To improve the performance of the G-ILC algorithm, the value of the learning gain β can be changed. Figure 9 shows the effect of β on the convergence rate.

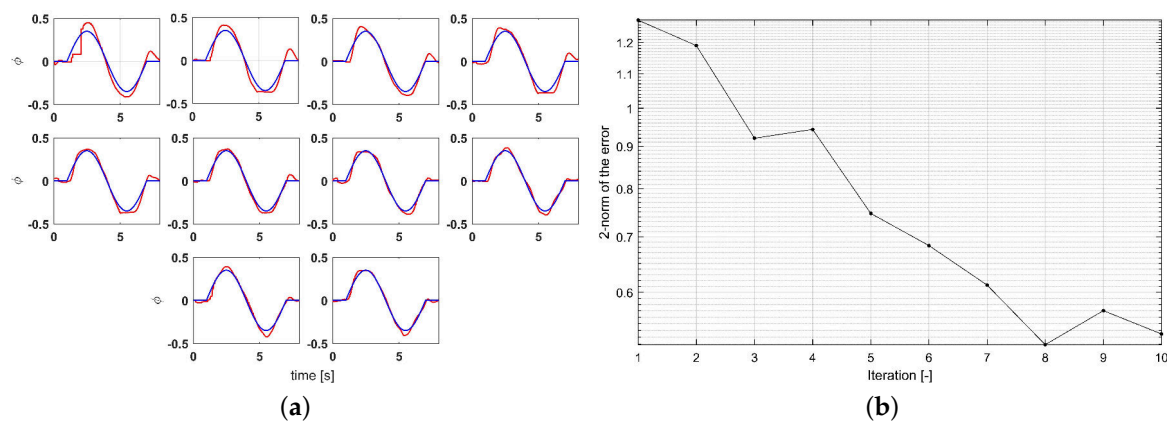


Figure 8. Experimental results of NO-ILC for different iterations with disturbance. (a) Experimental tracking with different iteration. (b) Monotonic convergence result.

Table 3. Simulation norm error results for attitude angles without disturbance.

ILC Approaches	Passes No.	$\ \hat{\theta} - \theta\ $	$\ \hat{\phi} - \phi\ $
G-ILC (without disturbance)	1	1.92	1.92
	3	1.38	1.38
	6	0.532	0.532
	16	0.309	0.303
NO-ILC (without disturbance)	1	1.92	1.92
	3	1.24	1.24
	6	0.476	0.476
	16	0.121	0.119

Table 4. Experimental norm error results for ILC algorithms with Disturbance Injection.

ILC Approaches	Trial (1)		Trial (3)		Trial (7)		Trial (10)	
	$\ e_{\theta}\ $	$\ e_{\psi}\ $	$\ e_{\theta}\ $	$\ e_{\psi}\ $	$\ e_{\theta}\ $	$\ e_{\psi}\ $	$\ e_{\theta}\ $	$\ e_{\psi}\ $
G-ILC + Disturbance Injection	1.277	2.834	0.920	1.281	0.612	0.705	0.574	0.534
NO-ILC + Disturbance Injection	1.283	3.214	0.926	1.212	0.562	0.633	0.434	0.446

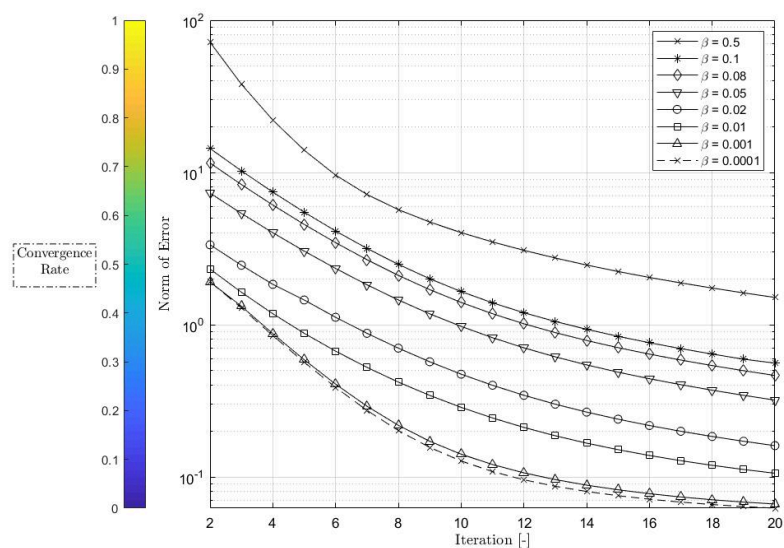


Figure 9. Optimized error convergence rate with a variation on the learning gain β values, as proposed in Equations (35) and (42).

For a large class of practical systems, such as UAV reference tracking (as required for power line surveillance and monitoring) it is required that the output achieves perfect tracking at more than one defined time and enables the system error to converge to zero norm as rapidly as possible. Consequently, it includes future work on an alternative controller (i.e., ILC with hybrid controller) as an extension to enhance the tracking performance at subset (instantaneous in time) for many critical positions.

6. Conclusions

The suggested G-ILC and NO-ILC have been formulated and applied to the problem of reference tracking for UAV. When comparing the findings, the NO-ILC has shown superior tracking performance. Furthermore, the suggested NO-ILC has shown substantially improvement over the G-ILC in terms of error decrease and monotonic convergence. The results of the simulations and experiments both with and without an external disturbance show the proposed ILC performance for the two methods. The results the potential potential to achieve good trajectory tracking.

The NO-ILC method could form the basis for a power line inspection system. The repetitive nature of the power line geometries lends itself to this approach. However there are many control challenges to be faced, such as disturbances in the form of steady wind and unsteady wind gusts, and decision-making in the face of uncertainty. This points to the urgent need for additional future work for expanding ILC (i.e., point-to-point with hybrid controller) for tracking identification, for instance, through a straight conductor for a electrical overhead conductors monitoring-task.

Author Contributions: Conceptualization, investigation: H.A.F.; methodology: H.A.F., P.L.; software, validation: H.A.F.; writing—original draft preparation: H.A.F.; writing—review and editing: P.L., J.W.; supervision: P.L., J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research is partly funded by the Jordan Government (Mutah University), and partly funded by the Cranfield University.

Acknowledgments: The authors would like to thank the anonymous reviewers for their useful suggestions, and support family members during the pandemic crisis.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

COG	Center Of Gravity
DOF	Degrees Of Freedom
G-ILC	Gradient-based Iterative Learning Control
GPS	Global Positioning System
ILC	Iterative Learning Control
IMU	Inertial Measurement Unit
LiPo	Lithium Polymer
LQG	Linear Quadratic Regulation
MIMO	Multi Input Multi Output
MIT	Massachusetts Institute of Technology
MPC	Model Predictive control
NO-ILC	Norm Optimal Iterative Learning Control
PD	Proportional Derivative
PID	Proportional Integral Derivative
SISO	Single Input Single Output
SMC	Sliding Mode Control
UAV	Unmanned Aerial Vehicle

References

1. Dehghanian, P.; Aslan, S.; Dehghanian, P. Maintaining electric system safety through an enhanced network resilience. *IEEE Trans. Ind. Appl.* **2018**, *54*, 4927–4937. [[CrossRef](#)]

2. Kogan, V.I.; Gursky, R.J. Transmission towers inventory. *IEEE Trans. Power Deliv.* **1996**, *11*, 1842–1852. [[CrossRef](#)]
3. Moradkhani, A.; Haghifam, M. R.; Mohammadzadeh, M. Failure rate modelling of electric distribution overhead lines considering preventive maintenance. *IET Gener. Transm. Distrib.* **2014**, *8*, 1028–1038. [[CrossRef](#)]
4. Na, H.J.; Yoo, S.J. PSO-based dynamic UAV positioning algorithm for sensing information acquisition in Wireless Sensor Networks. *IEEE Access* **2019**, *7*, 77499–77513. [[CrossRef](#)]
5. Davis, E.; Pounds, P.E. Direct sensing of thrust and velocity for a quadrotor rotor array. *IEEE Robot. Autom. Lett.* **2017**, *2*, 1360–1366. [[CrossRef](#)]
6. Zhong, M.; Cao, Q.; Guo, J.; Zhou, D. Simultaneous lever-arm compensation and disturbance attenuation of POS for a UAV surveying system. *IEEE Trans. Instrum. Meas.* **2016**, *65*, 2828–2839. [[CrossRef](#)]
7. Wang, Q.; Xiong, H.; Qiu, B. The Attitude Control of Transmission Line Fault Inspection UAV Based on ADRC. In Proceedings of the International Conference on Industrial Informatics-Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII), Nicosia, Cyprus, 27–29 September 2017; pp. 186–189.
8. Menéndez, O.; Pérez, M.; Auat Cheein, F. Visual-based positioning of aerial maintenance platforms on overhead transmission lines. *Appl. Sci.* **2019**, *9*, 165. [[CrossRef](#)]
9. Jones, D. Power line inspection-a UAV concept. In Proceedings of the IEE Forum on Autonomous Systems, London, UK, 28 November 2005; p. 8.
10. Bouabdallah, S.; Murrieri, P.; Siegwart, R. Design and control of an indoor micro quadrotor. *IEEE Int. Conf. Robot. Autom.* **2004**, *5*, 4393–4398.
11. Ren, J.; Liu, D.X.; Li, K.; Liu, J.; Feng, Y.; Lin, X. Cascade PID controller for quadrotor. In Proceedings of the 2016 IEEE International Conference on Information and Automation (ICIA), Ningbo, China, 1–3 August 2016; pp. 120–124.
12. Bouabdallah, S.; Noth, A.; Siegwart, R. PID vs. LQ control techniques applied to an indoor micro quadrotor. *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)* **2004**, *3*, 2451–2456.
13. Shaik, M.K.; Whidborne, J.F. Robust sliding mode control of a quadrotor. In Proceedings of the 2016 UKACC 11th International Conference on Control (CONTROL), Belfast, UK, 31 August–2 September 2016.
14. Tan, L.; Lu, L.; Jin, G. Attitude stabilization control of a quadrotor helicopter using integral backstepping. In Proceedings of the International Conference on Automatic Control and Artificial Intelligence, Xiamen, China, 24–26 March 2012; pp. 573–577.
15. Nandong, J. A unified design for feedback-feedforward control system to improve regulatory control performance. *Int. J. Control Autom. Syst.* **2015**, *13*, 91–98. [[CrossRef](#)]
16. Ahn, H.S.; Chen, Y.; Moore, K.L. Iterative learning control: Brief survey and categorization. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **2007**, *37*, 1099–1121. [[CrossRef](#)]
17. Ke, C.; Ren, J.; Quan, Q. Saturated D-type ILC for Multicopter Trajectory Tracking Based on Additive State Decomposition. In Proceedings of the IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS), Enshi, China, 25–27 May 2018; pp. 1146–1151.
18. Dong, J.; He, B. Novel fuzzy PID-type iterative learning control for quadrotor UAV. *Sensors* **2019**, *19*, 24. [[CrossRef](#)] [[PubMed](#)]
19. Koçan, O.; Manecy, A.; Poussot-Vassal, C. A Practical Method to Speed-up the Experimental Procedure of Iterative Learning Controllers. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October, 2018; pp. 6411–6416.
20. Li, X.; Ren, Q.; Xu, J.X. Precise Speed Tracking Control of a Robotic Fish Via Iterative Learning Control. *IEEE Trans. Ind. Electron.* **2016**, *63*, 2221–2228. [[CrossRef](#)]
21. Meng, D.; Moore, K.L. Learning to cooperate: Networks of formation agents with switching topologies. *Automatica* **2016**, *64*, 278–293. [[CrossRef](#)]
22. Chen, T.; Shan, J. Distributed control of multiple flexible manipulators with unknown disturbances and dead-zone input. *IEEE Trans. Ind. Electron.* **2019**. [[CrossRef](#)]
23. Bristow, D.A.; Tharayil, M.; Alleyne, A.G. A survey of iterative learning control. *IEEE Control Syst. Mag.* **2006**, *26*, 96–114.
24. Moore, K.L. *Iterative Learning Control for Deterministic Systems*; Springer: Berlin/Heidelberg, Germany, 2012.
25. Gao, F.; Chen, W.; Li, Z.; Li, J.; Xu, B. Neural Network-Based Distributed Cooperative Learning Control for Multiagent Systems via Event-Triggered Communication. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 407–419. [[CrossRef](#)]

26. He, W.; Meng, T.; He, X.; Sun, C. Iterative Learning Control for a Flapping Wing Micro Aerial Vehicle under Distributed Disturbances. *IEEE Trans. Cybern.* **2019**, *49*, 1524–1535. [\[CrossRef\]](#)
27. Ren, J.; Quan, Q.; Liu, C.; Cai, K.-Y. Docking control for probe-drogue refueling: An additive-state-decomposition-based output feedback iterative learning control method. *Chin. J. Aeronaut.* **2019**, *33*, 1016–1025. [\[CrossRef\]](#)
28. Hock, A.; Schoellig, A.P. Distributed iterative learning control for a team of quadrotors. In Proceedings of the 2016 IEEE 55th Conference on Decision and Control (CDC), Las Vegas, Nevada, USA, 12–14 December 2016; pp. 4640–4646.
29. Pipatpaibul, P.I.; Ouyang, P.R. Application of online iterative learning tracking control for quadrotor UAVs. *ISRN Robot.* **2013**. [\[CrossRef\]](#)
30. Zhaowei, M.; Tianjiang, H.; Lincheng, S.; Weiwei, K.; Boxin, Z.; Kaidi, Y. An iterative learning controller for quadrotor UAV path following at a constant altitude. In Proceedings of the 2015 34th Chinese Control Conference (CCC), Hangzhou, China, 28–30 July 2015; pp. 4406–4411.
31. Giernacki, W. Iterative learning method for in-flight auto-tuning of UAV controllers based on basic sensory information. *Appl. Sci.* **2019**, *9*, 648. [\[CrossRef\]](#)
32. Schoellig, A.P.; Mueller, F.L.; D’Andrea, R. Optimization-based iterative learning for precise quadcopter trajectory tracking. *Auton. Robots* **2012**, *33*, 103–127. [\[CrossRef\]](#)
33. Devasia, S. Iterative Machine Learning for Output Tracking. *IEEE Trans. Control Syst. Technol.* **2019**, *27*, 516–526. [\[CrossRef\]](#)
34. Foudeh, H.A.; Luk, P.; Whidborne, J.F. Quadrotor System Design for a 3 DOF platform based on Iterative Learning Control. In Proceedings of the 2019 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED UAS), Cranfield, UK, 25–27 November 2019; pp. 53–59.
35. Altin, B.; Wang, Z.; Hoelzle, D.J.; Barton, K. Robust Monotonically Convergent Spatial Iterative Learning Control: Interval Systems Analysis via Discrete Fourier Transform. *IEEE Trans. Control Syst. Technol.* **2019**, *27*, 2470–2483. [\[CrossRef\]](#)
36. Wang, Y.; Dassau, E.; Doyle, F.J., III Closed-loop control of artificial pancreatic β -cell in type 1 diabetes mellitus using model predictive iterative learning control. *IEEE Trans. Biomed. Eng.* **2009**, *57*, 211–219. [\[CrossRef\]](#) [\[PubMed\]](#)
37. Barton, K.; Van De Wijdeven, J.; Alleyne, A.; Bosgra, O.; Steinbuch, M. Norm optimal cross-coupled iterative learning control. In Proceedings of the IEEE Conference on Decision and Control, Cancun, Mexico, 9–11 December 2008; pp. 3020–3025.
38. Yang, X.; Ruan, X. Reinforced gradient-type iterative learning control for discrete linear time-invariant systems with parameters uncertainties and external noises. *IMA J. Math. Control. Inf.* **2017**, *34*, 1117–1133. [\[CrossRef\]](#)
39. Gu, P.; Tian, S.; Chen, Y. Iterative Learning Control Based on Nesterov Accelerated Gradient Method. *IEEE Access* **2019**, *7*, 115836–115842. [\[CrossRef\]](#)
40. Liu, S.; Wu, T.J. Robust Iterative Learning Control Design Based on Gradient Method. *IFAC Proc. Vol.* **2004**, *37*, 613–618. [\[CrossRef\]](#)
41. Li, X.; Ruan, X.; Liu, Y. Learning-Gain-Adaptive Iterative Learning Control to Linear Discrete-Time-Invariant Systems. *IEEE Access* **2019**, *7*, 98934–98945. [\[CrossRef\]](#)
42. Lv, Y.; Chi, R. Data-driven adaptive iterative learning predictive control. In Proceedings of the 2017 6th Data Driven Control and Learning Systems (DDCLS), Chongqing, China, 26–27 May 2017; pp. 374–377.
43. Ratcliffe, Y.; Ruan, X.; Li, X. Optimized Iterative Learning Control for Linear Discrete-Time-Invariant Systems. *IEEE Access* **2019**, *7*, 75378–75388.
44. Karwoski, K. *Quadrocopter Control Design and Flight Operation*; NASA USRP—Internship Final Report; Marshall Space Flight Center: Huntsville, AL, USA, 2011.

